



# Configure MACSec

This module describes how to configure Media Access Control Security (MACSec) encryption on the ASR 9000 Series Aggregation Services Routers. MACSec is a Layer 2 IEEE 802.1AE standard for encrypting packets between two MACSec-capable routers.

## Feature History for Configure MACSec

Release	Modification
Release 5.3.2	This feature was introduced.
Release 6.0.1	This feature was modified to support VLAN sub-interfaces and bundles.
Release 6.1.2	This feature was modified to introduce MACsec as a service.
Release 6.3.3	Introduced the support for global MACsec shutdown.
Release 6.3.3	Introduced the support for MACsec SAK rekey interval.
Release 7.1.3	MACSec support was introduced on Cisco ASR 9000 5th generation line cards, Cisco ASR 9903 1.6T chassis and Cisco ASR 9903 2T port expansion card running Cisco IOS XR 64-bit.

- [Understanding MACsec Encryption, on page 2](#)
- [Advantages of Using MACsec Encryption, on page 3](#)
- [Types of MACsec Implementation, on page 3](#)
- [MKA Authentication Process, on page 4](#)
- [MACSec Support on Line Cards and Routers, on page 5](#)
- [MACsec PSK, on page 5](#)
- [Fallback PSK, on page 6](#)
- [Configuring and Verifying MACSec Encryption , on page 6](#)
- [Configuring and Verifying MACsec Encryption as a Service, on page 26](#)
- [Global MACsec Shutdown, on page 43](#)

# Understanding MACsec Encryption

Security breaches can occur at any layer of the OSI model. At Layer 2, some of the common breaches at Layer 2 are MAC address spoofing, ARP spoofing, Denial of Service (DoS) attacks against a DHCP server, and VLAN hopping.

MACsec secures data on physical media, making it impossible for data to be compromised at higher layers. As a result, MACsec encryption takes priority over any other encryption method such as IPsec and SSL, at higher layers. MACsec is configured on Customer Edge (CE) router interfaces that connect to Provider Edge (PE) routers and on all the provider router interfaces.

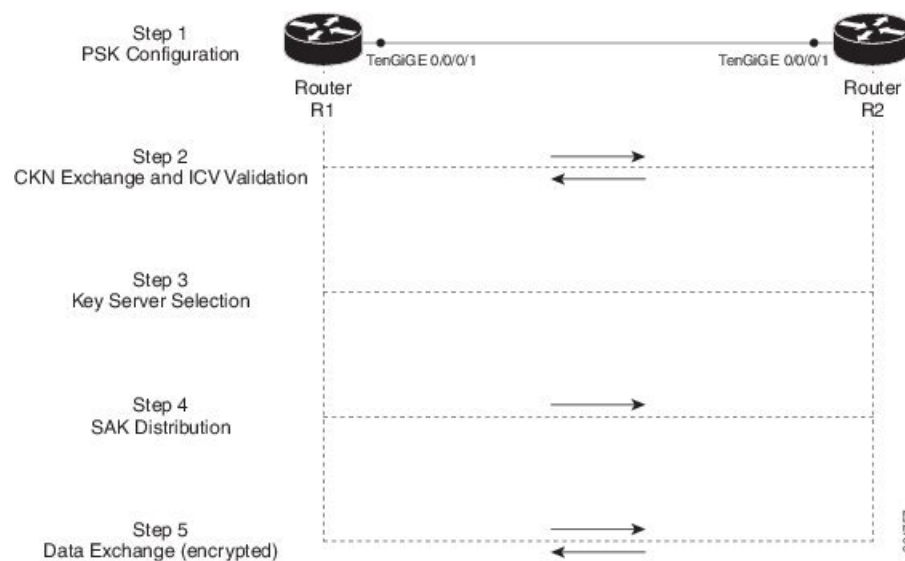
MACservice can be deployed in the network as a technology or as a service. For more information, see [Types of MACsec Implementation, on page 3](#)

## MACsec Authentication Process

MACsec provides encryption using Advanced Encryption Standard (AES) algorithm at the Layer 2. MACsec uses the MACsec Key Agreement protocol (MKA) to exchange session keys, and manage encryption keys.

The MACsec encryption process is illustrated in the following figure and description.

**Figure 1: MACsec Encryption Process**



**Step 1:** When a link is first established between two routers, they become peers. Mutual peer authentication takes place by configuring a Pre-shared Key (PSK).

**Step 2:** On successful peer authentication, a connectivity association is formed between the peers, and a secure Connectivity Association Key Name (CKN) is exchanged. After the exchange, the MKA ICV is validated with a Connectivity Association Key (CAK), which is effectively a secret key.

**Step 3:** A key server is selected between the routers, based on the configured key server priority. Lower the priority value, higher the preference for the router to become the key server. If no value is configured, the default value of 16 is taken to be the key server priority value for the router. Lowest priority value configures that router as the key server, while the other router functions as a key client. The following rules apply to key server selection:

- Numerically lower values of key server priority and SCI are accorded the highest preference.
- Each router selects a peer advertising the highest preference as its key server provided that peer has not selected another router as its key server or is not willing to function as the key server.
- In the event of a tie for highest preferred key server, the router with the highest priority SCI is chosen as key server (KS).

**Step 4:** A security association is formed between the peers. The key server generates and distributes the Secure Association Key (SAK) to the key client (peer). SAKs are generated for every data exchange between the peers.

**Step 5:** Encrypted data is exchanged between the peers.

## Advantages of Using MACsec Encryption

- **Client-Oriented Mode:** MACsec is used in setups where two routers that are peering with each other can alternate as a key server or a key client prior to exchanging keys. The key server generates and maintains the CAK between the two peers.
- **Data Integrity Check:** MACsec uses MKA to generate an Integrity Check Value (ICV) for the frame arriving on the port. If the generated ICV is the same as the ICV in the frame, then the frame is accepted; otherwise it is dropped.
- **Data Encryption:** MACsec provides port-level encryption on the line card of the router. This means that the frames sent out of the configured port are encrypted and frames received on the port are decrypted. MACsec also provides a mechanism where you can configure whether only encrypted frames or all frames (encrypted and plain) are accepted on the interface.
- **Replay Protection:** When frames are transmitted through the network, there is a strong possibility of frames getting out of the ordered sequence. MACsec provides a configurable window that accepts a specified number of out-of-sequence frames.
- **Support for Clear Traffic:** If configured accordingly, data that is not encrypted is allowed to transit through the port.

## Types of MACsec Implementation

MACsec is implemented in the following ways:

- **MACsec** where it serves as an encryption method for all traffic on Ethernet links.

For more information on configuring MACsec, see *Creating a MACsec Keychain* and *Creating a MACsec Policy*

- **MACsec as a service** where it serves as an encryption method for L2VPN and L3VPN traffic over a provider network. It provides a mechanism to provide encryption or decryption service for selected traffic across the WAN core. For example: a service provider can charge encryption of voice calls at a premium. This solution supports both Point-to-Point as well as Multipoint service for all the traffic on the network.

For more information on configuring MACsec as a service, see [Configuring MACsec as a Service, on page 27](#)

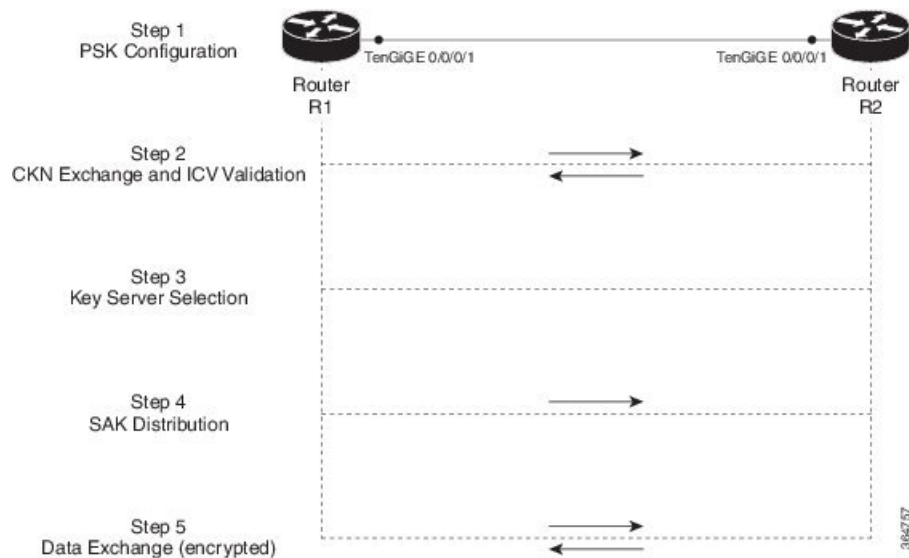
Both MACsec and MACsec service are mutually exclusive and can be deployed in the same network.

# MKA Authentication Process

MACsec provides encryption at the Layer 2, which is provided by the Advanced Encryption Standard (AES) algorithm that replaces the DES algorithm. MACsec uses the MACsec Key Agreement protocol (MKA) to exchange session keys, and manage encryption keys.

The MACsec encryption process is illustrated in the following figure and description.

**Figure 2: MKA Encryption Process**



**Step 1:** When a link is first established between two routers, they become peers. Mutual peer authentication takes place by configuring a Pre-shared Key (PSK).

**Step 2:** On successful peer authentication, a connectivity association is formed between the peers, and a secure Connectivity Association Key Name (CKN) is exchanged. After the exchange, the MKA ICV is validated with a Connectivity Association Key (CAK), which is effectively a secret key.

**Step 3:** A key server is selected between the routers, based on the configured key server priority. Lower the priority value, higher the preference for the router to become the key server. If no value is configured, the default value of 16 is taken to be the key server priority value for the router. Lowest priority value configures that router as the key server, while the other router functions as a key client. The following rules apply to key server selection:

- Numerically lower values of key server priority and SCI are accorded the highest preference.
- Each router selects a peer advertising the highest preference as its key server provided that peer has not selected another router as its key server or is not willing to function as the key server.
- In the event of a tie for highest preferred key server, the router with the highest priority SCI is chosen as key server (KS).

**Step 4:** A security association is formed between the peers. The key server generates and distributes the Secure Association Key (SAK) to the key client (peer). Each secure channel is supported by an overlapped sequence of Security Associations(SA). Each SA uses a new Secure Association Key (SAK).

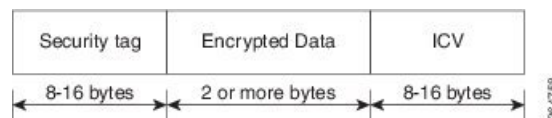
**Step 5:** Encrypted data is exchanged between the peers.

### MACsec Frame Format

The MACsec header in a frame consists of three components as illustrated in the following figure.

- **Security tag:** The security tag is 8-16 bytes in length and identifies the SAK to be used for the frame. With Secure Channel Identifier (SCI) encoding, the security tag is 16 bytes in length, and without the encoding, 8 bytes in length (SCI encoding is optional). The security tag also provides replay protection when frames are received out of sequence.
- **Secure data:** This is the data in the frame that is encrypted using MACsec and can be 2 or more octets in length.
- **ICV:** The ICV provides the integrity check for the frame and is usually 8-16 bytes in length, depending on the cipher suite. Frames that do not match the expected ICV are dropped at the port.

**Figure 3: MACsec Frame Format**



## MACSec Support on Line Cards and Routers

MACSec is supported on the following line cards and routers:

- The MACSec technology is supported only on the 200G and 400G modular line cards when used with the following Modular Port Adapters:
  - 20-port 10 Gigabit Modular Port Adapter (A9K-MPA-20X10GE)
  - 1-port 100 Gigabit Modular Port Adapter (A9K-MPA-1X100GE)
  - 2-port 100 Gigabit Modular Port Adapter (A9K-MPA-2X100GE)
- Cisco ASR 9000 Series 400-Gbps IPoDWDM Line Card (A9K-400G-DWDM-TR)
- 4X100 GE and 8X100 GE OTN Line Card



#### Note

- MACSec is not supported on ASR9000 24-port dual-rate 10G/1G service edge-optimized line card (A9K-24X10GE-1G-SE).

## MACsec PSK

A pre-shared key includes a connectivity association key name (CKN) and a connectivity association key (CAK). A pre-shared key is exchanged between two devices at each end of a point-to-point (P2P) link to enable MACsec using static CAK security mode. The MACsec Key Agreement (MKA) protocol is enabled after the pre-shared keys are successfully verified and exchanged. The pre-shared keys, the CKN and CAK, must match on both ends of a link.

## Fallback PSK

Fallback is a session recovery mechanism when primary PSK fails to bring up secured MKA session. It ensures that a PSK is always available to perform MACsec encryption and decryption.

- In CAK rollover of primary keys, if latest active keys are mismatched, system performs a hitless rollover from current active key to fallback key, provided the fallback keys match.
- If a session is up with fallback, and primary latest active key configuration mismatches are rectified between peers, system performs a hitless rollover from fallback to primary latest active key.



---

**Note**

- A valid Fallback PSK (CKN and CAK) must be configured with infinite lifetime. If the fallback PSK is configured with CAK mismatch, the only recovery mechanism is to push a new set of PSK configurations (both on fallback PSK keychain and primary PSK chain in that order) on all the association members.
- In P2P topologies, a rollover to the fallback PSK happens when either of the nodes in the Secure Association (SA) cannot peer up with the primary PSK. Whereas, in P2MP, the fallback happens only at the expiry or deletion of the primary key on all peers, not just on one of the peers. On deletion or expiry of the primary PSK on one of the nodes, say R1, a new key server is chosen among the peer nodes that does a SAK rekey for the remaining nodes. This ensures that R1 is no longer part of the SA, and the network drops all traffic to and from R1.

---

The following is a sample syslog for session secured with fallback PSK:

```
%L2-MKA-5-SESSION_SECURED_WITH_FALLBACK_PSK : (Hu0/1/0/0) MKA session secured, CKN:ABCD
```

For more information on MACsec fallback PSK configuration, see [Applying MACsec Configuration on an Interface, on page 15](#).

## Configuring and Verifying MACSec Encryption

MACSec can be configured on physical ethernet interfaces, VLAN sub-interfaces, or interface bundles (link bundles), as explained in this section.



---

**Note**

MACSec on a VLAN sub-interface is configured in same way as on a physical interface. For a successful MKA session to be up on any VLAN sub-interface, it must have a valid tagging protocol encapsulation and VLAN identifier assigned. All Ethernet sub-interfaces always default to the 802.1Q VLAN encapsulation. However, the VLAN identifier must be explicitly defined. The sub-interfaces belonging to a physical interface can have the following encapsulation combinations:

- 802.1Q with a single tag
  - 802.1Q with double tags
  - 802.1ad with a single tag
  - 802.1ad with double tags
-

**Use Case 1: MACSec in a L2VPN**

The following figure illustrates the use of MACSec in a L2VPN network. In this topology, MACSec is configured on the PE-facing interfaces of the CE routers. The interfaces can be physical ethernet interfaces or VLAN sub-interfaces.

In a L2VPN network that uses an Ethernet over MPLS (EoMPLS) pseudowire, the traffic between CE routers is encrypted by MACSec with VLAN tags in clear. The following figure illustrates the use of MACSec in a L2VPN cloud using an EoMPLS pseudowire. MACSec is configured on the PE-facing VLAN sub-interfaces of the CE router. The PE router encapsulates the MACSec frames with VLAN tags and MPLS labels in clear and sends the frames over the EoMPLS pseudowire.

The following table lists the number of sub-interfaces with MACSec supported in a L2VPN.

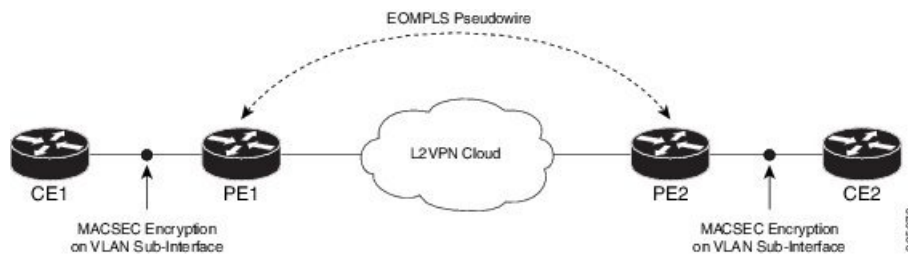


**Note** To achieve scaling, sub-interfaces must be used.

**Table 1: Supported MACSec Sessions on Sub-Interfaces**

Interface Type	No. of Supported MACSec sessions (P2P)
10-GigE	5
40-GigE	21
100-GigE	42

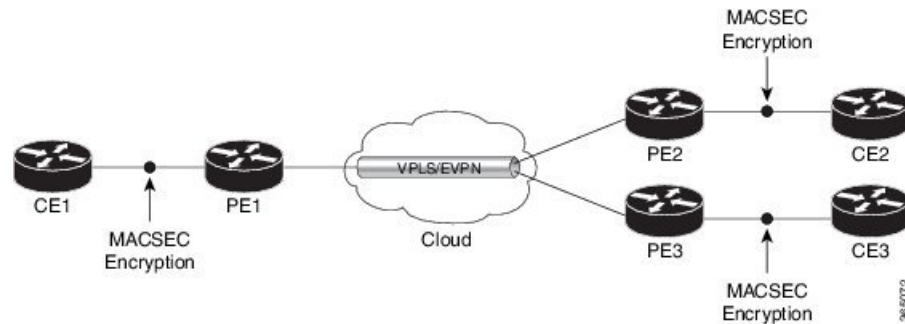
**Figure 4: MACSec in a L2VPN Cloud**



**Use Case 2: MACSec in a VPLS/EVPN**

A typical VPLS network often suffers the injection of labeled traffic from potential hackers. The following figure illustrates the use of MACSec in a VPLS/EVPN network for encrypting the data being exchanged over the VPLS cloud. In this topology MACSec is configured on the PE-facing interfaces of the CE routers. The interfaces can be physical ethernet interfaces or VLAN sub-interfaces.

Figure 5: MACSec in a VPLS/EVPN Cloud



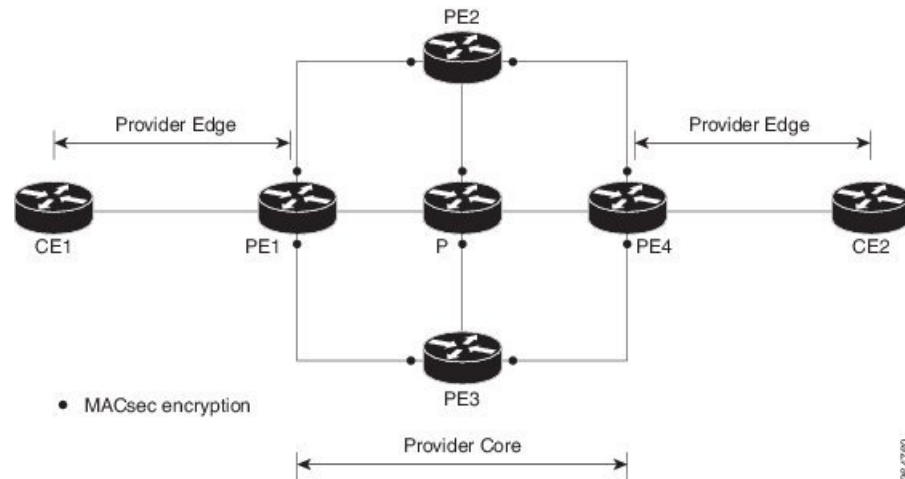
**Use Case 3: MACSec in an MPLS Core Network**

MACSec in an MPLS core network can be configured on physical interfaces, sub-interfaces or link bundles (Link Aggregation Group or LAG).

In the following topology, MACSec is configured on all router links in the MPLS core. This deployment is useful when the MPLS network spans data centers that are not co-located in the same geography. Each link is, therefore, a link between two data centers and all data exchanged is encrypted using MACSec.

The following figure illustrates the use of MACSec on physical interfaces in an MPLS core network.

Figure 6: MACSec on Physical Interfaces in an MPLS Core Network

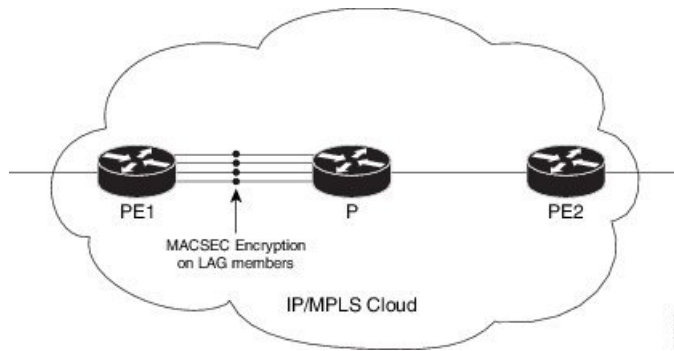


When MACSec is configured on the members of a LAG, an MKA session is set up for each member. SAK is exchanged for each LAG member and encryption/decryption takes place independently of other members in the group. MACSec can also be configured on VLAN sub-interfaces in these networks.

The following figure illustrates the use of MACSec on a link bundle in an MPLS core network.



Figure 7: MACSec on a Link Bundle in an MPLS Core Network



The following section describes procedures for configuring and verifying MACSec configuration in any of the described deployment modes.

Prior to configuring MACSec on a router interface, the MACSec key chain and MACSec policy must be defined. Configuring MACSec encryption involves the following steps:

1. Creating a MACSec Key Chain
2. Creating a MACSec Policy
3. Applying MACSec on a Physical Interface

## Creating a MACsec Key Chain

A MACsec keychain is a collection of keys used to authenticate peers needing to exchange encrypted information. While creating a keychain, we define the key(s), key string with password, the cryptographic algorithm, and the key lifetime.

MACsec Keychain Keyword	Description
Key	The MACsec key or the CKN can be up to 64 characters in length. The key must be of an even number of characters. Entering an odd number of characters will exit the MACsec configuration mode.
Key-string	The MACsec key-string or the CAK can be either 32 characters or 64 characters in length (32 for AES-128, 64 for AES-256).
Lifetime	This field specifies the validity period of a key. It includes a start time, and an expiry time. We recommend you to set the value for expiry time as <i>infinite</i> .

### Guidelines for Configuring MACsec Keychain

MACsec keychain management has the following configuration guidelines:

- To establish MKA session, ensure that the MACsec key (CKN) and key-string (CAK) match at both ends.

- MKA protocol uses the latest active key available in the Keychain. This key has the latest Start Time from the existing set of currently active keys. You can verify the values using the **show key chain** *keychain-name* command.
- Deletion or expiry of current active key brings down the MKA session resulting in traffic hit. We recommend you to configure the keys with infinite lifetime. If fallback is configured, traffic is safeguarded using fallback on expiry or deletion of primary-keychain active key.
- To achieve successful key rollover (CAK-rollover), the new key should be configured such that it is the latest active key, and kicks-in before the current key expires.
- We recommend an overlap of at least one minute for hitless CAK rollover from current key to new key.
- Start time and Expiry time can be configured with future time stamps, which allows bulk configuration for daily CAK rotation without any intervention of management agent.

## SUMMARY STEPS

1. Enter the global configuration mode and provide a name for the MACsec keychain; for example, mac\_chain.
2. Enter the MACsec mode.
3. Provide a name for the MACsec key.
4. Enter the key string and the cryptographic algorithm to be used for the key.
5. Enter the validity period for the MACsec key (CKN) also known as the lifetime period.
6. Commit your configuration.

## DETAILED STEPS

**Step 1** Enter the global configuration mode and provide a name for the MACsec keychain; for example, mac\_chain.

**Example:**

```
RP/0/RSP0/CPU0:router(config)#key chain mac_chain
```

**Step 2** Enter the MACsec mode.

**Example:**

```
RP/0/RSP0/CPU0:router(config-mac_chain)#macsec
```

**Step 3** Provide a name for the MACsec key.

The key can be up to 64 characters in length. The key must be of an even number of characters. Entering an odd number of characters will exit the MACsec configuration mode.

**Example:**

```
RP/0/RSP0/CPU0:router(config-mac_chain-MacSec)#key 1234abcd5678
```

You can also configure a fall-back pre-shared key (PSK) to ensure that a PSK is always available to perform MACsec encryption and decryption. The fallback PSK along with the primary PSK ensures that the session remains active even if the primary PSK is mismatched or there is no active key for the primary PSK.

The configured key is the CKN that is exchanged between the peers.

**Note** If you are configuring MACsec to inter-operate with a MACsec server that is running software prior to Cisco IOS XR Release 6.1.3, then ensure that the MACsec key length is of 64 characters. You can add extra zero characters to the MACsec key so that the length of 64 characters is achieved. If the key length is lesser than 64 characters, authentication will fail.

**Step 4** Enter the key string and the cryptographic algorithm to be used for the key.

**Example:**

The key string is the CAK that is used for ICV validation by the MKA protocol.

**! For AES 128-bit encryption**

```
RP/0/RSP0/CPU0:router (config-mac_chain-MacSec-1234abcd5678) #key-string 12345678123456781234567812345678
cryptographic-algorithm AES-128-CMAC
```

**! For AES 256-bit encryption**

```
RP/0/RSP0/CPU0:router (config-mac_chain-MacSec-1234abcd5678) #key-string
1234567812345678123456781234567812345678123456781234567812345678 cryptographic
-algorithm AES-256-CMAC
```

**Note** In this example, we have used the AES 256-bit encryption algorithm, and therefore, the key string is 64 hexadecimal characters in length. A 256-bit encryption algorithm uses a larger key that requires more rounds of hacking to be cracked. 256-bit algorithms provide better security against large mass security attacks, and include the security provided by 128-bit algorithms.

**Step 5** Enter the validity period for the MACsec key (CKN) also known as the lifetime period.

The lifetime period can be configured, with a duration in seconds, as a validity period between two dates (for example, Jan 01 2014 to Dec 31 2014), or with infinite validity.

The key is valid from the time you configure (in HH:MM:SS format). Duration is configured in seconds.

**Example:**

```
RP/0/RSP0/CPU0:router (config-mac_chain-MacSec-1234abcd5678) #lifetime 05:00:00 01
January 2015 duration 1800
```

An example of configuring the lifetime for a defined period:

```
RP/0/RSP0/CPU0:router (config-mac_chain-MacSec-1234abcd5678) #lifetime 05:00:00 20
february 2015 12:00:00 30 september 2015
```

An example of configuring the lifetime as infinite:

```
RP/0/RSP0/CPU0:router (config-mac_chain-MacSec-1234abcd5678) #lifetime
05:00:00 01 January 2015 infinite
```

**Note** When a key has expired, the MACsec session is torn down and running the **show macsec mka session** command does not display any information. If you run the **show macsec mka interface detail** command, the output displays **\*\*\* No Active Keys Present \*\*\*** in the PSK information.

**Step 6** Commit your configuration.

**Example:**

```
RP/0/RSP0/CPU0:router(config-mac_chain-MacSec-1234abcd5678)#commit
```

---

This completes the configuration of the MACsec keychain.

## Prerequisites for Configuring MACSec on Bundle Member Interfaces

To enable MACSec on bundle members, an user-defined policy must be configured with Should-Secure policy, or Must-Secure policy with **policy-exception LACP-in-clear** command.




---

**Note** By default, the system uses the Must-Secure security policy.

---

**Example: Configuring MACSec on Bundle Member With Should-Secure Policy**

```
(config)#macsec-policy should-secure
(config-macsec-policy)#security-policy should-secure
(config-macsec-policy)#commit
```

```
sh runn macsec-policy should-secure
macsec-policy should-secure
security-policy should-secure
!
```

```
router(config)# interface HundredGigE 0/1/1/1 # Applying the Should-Secure MACSec Policy
on Bundle Member Interface
router(config-if)# bundle id 12 mode active
router(config-if)# macsec psk-keychain kcl policy should-secure
```

**Example: Configuring MACSec on Bundle Member With Must-Secure Policy**

```
(config)#macsec-policy must-secure
(config-macsec-policy)#security-policy must-secure
(config-macsec-policy)#policy-exception lacp-in-clear
(config-macsec-policy)#commit
```

```
#sh runn macsec-policy must-secure
macsec-policy must-secure
security-policy must-secure
policy-exception lacp-in-clear
!
```

```
router(config)# interface HundredGigE 0/1/1/2 #Applying the Must-Secure MACSec Policy on
Bundle Member Interface
router(config-if)# bundle id 12 mode active
router(config-if)# macsec psk-keychain kcl policy must-secure
```

# Creating a User-Defined MACsec Policy

## SUMMARY STEPS

1. Enter the global configuration mode, and enter a name (`mac_policy`) for the MACsec policy.
2. Configure the cipher suite to be used for MACsec encryption.
3. Configure the confidentiality offset for MACsec encryption.
4. Enter the key server priority.
5. Configure the security policy parameters, either Must-Secure or Should-Secure.
6. Configure the replay protection window size.
7. Configure the ICV for the frame arriving on the port.
8. Commit your configuration and exit the global configuration mode.
9. Confirm the MACsec policy configuration.

## DETAILED STEPS

**Step 1** Enter the global configuration mode, and enter a name (`mac_policy`) for the MACsec policy.

**Example:**

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# macsec-policy mac_policy
```

**Step 2** Configure the cipher suite to be used for MACsec encryption.

**Example:**

```
RP/0/RSP0/CPU0:router(config-mac_policy)# cipher-suite GCM-AES-XPN-256
RP/0/RSP0/CPU0:router(config-mac_policy)#GCM-AES-128
GCM-AES-256
GCM-AES-XPN-128
GCM-AES-XPN-256
```

**Note** In this example, we have used the GCM-AES-XPN-256 encryption algorithm. A 256-bit encryption algorithm uses a larger key that requires more rounds of hacking to be cracked. 256-bit algorithms provide better security against large mass security attacks, and include the security provided by 128-bit algorithms. Extended Packet Numbering (XPN) is used to reduce the number of key rollovers while data is sent over high speed links. It is therefore highly recommended to use GCM-AES-XPN-256 encryption algorithm for higher data ports.

**Step 3** Configure the confidentiality offset for MACsec encryption.

**Example:**

```
RP/0/RSP0/CPU0:router(config-mac_policy)# conf-offset CONF-OFFSET-30
```

**Step 4** Enter the key server priority.

You can enter a value between 0-255. Lower the value, higher the preference to be selected as the key server.

In this example, a value of 0 configures the router as the key server, while the other router functions as a key client. The key server generates and maintains the SAK between the two routers. The default key server priority value is 16.

**Example:**

```
RP/0/RSP0/CPU0:router(config-mac_policy)# key-server-priority 0
```

**Step 5** Configure the security policy parameters, either Must-Secure or Should-Secure.

**Must-Secure:** Must-Secure imposes only MACsec encrypted traffic to flow. Hence, until MKA session is not secured, traffic will be dropped.

**Example:**

```
RP/0/RSP0/CPU0:router(config-mac_policy)# security-policy must-secure
```

**Should-Secure:** Should-Secure allows unencrypted traffic to flow until MKA session is secured. After the MKA session is secured, Should-Secure policy imposes only encrypted traffic to flow.

**Example:**

```
RP/0/RSP0/CPU0:router(config-mac_policy)# security-policy should-secure
```

**Table 2: MACsec Security Policies**

MKA		Secured MKA Session	Unsecured MKA Session
Security Policy	Must-secure	Encrypted traffic	Traffic drop (no Tx and no Rx)
	Should-secure	Encrypted traffic	Plain text or unencrypted traffic

**Step 6** Configure the replay protection window size.

**Example:**

```
RP/0/RSP0/CPU0:router(config-mac_policy)# window-size 64
```

This dictates the maximum out-of-sequence frames that are accepted. You can configure a value between 0 and 1024.

**Step 7** Configure the ICV for the frame arriving on the port.

**Example:**

```
RP/0/RSP0/CPU0:router(config-mac_policy)# include-icv-indicator
```

This parameter configures inclusion of the optional ICV Indicator as part of the transmitted MACsec Key Agreement PDU (MKPDU). This configuration is necessary for MACsec to interoperate with routers that run software prior to IOS XR version 6.1.3. This configuration is also important in a service provider WAN setup where MACsec interoperates with other vendor MACsec implementations that expect ICV indicator to be present in the MKPDU.

**Step 8** Commit your configuration and exit the global configuration mode.

**Example:**

```
RP/0/RSP0/CPU0:router(config-mac_policy)# exit
RP/0/RSP0/CPU0:router(config)# commit
RP/0/RSP0/CPU0:router(config)# exit
```

**Step 9** Confirm the MACsec policy configuration.

**Example:**

```
RP/0/RSP0/CPU0:router# show running-config macsec-policy

macsec-policy mac_policy
conf-offset CONF-OFFSET-30
security-policy must-secure
window-size 64
```

```
cipher-suite GCM-AES-XPN-256
key-server-priority 0
include-icv-indicator
```

---

This completes the configuration of the MACsec policy.

**Note**

- Small packets might be dropped when Data Delay Protection (DDP) is enabled on many MACsec enabled interfaces of a scaled setup. To avoid this, enable DDP only on the interfaces which are absolutely necessary.
- For Cisco ASR 9000 Series Routers to interoperate with Cisco ASR9000 Series Routers that are older than Release 6.2.3, configure a user defined MACsec policy with the `policy-exception lacp-in-clear` command to bring up the MKA sessions over bundle interfaces running in LACP modes.

## MACsec SAK Rekey Interval

From Cisco IOS XR Software Release 6.3.3 and later, you can set a timer value to rekey the MACsec secure association key (SAK) at a specified interval. This periodic refresh of SAK ensures that data encryption key is frequently updated. The configuration is effective on the node acting as a key server.

To set the rekey interval, use the **sak-rekey-interval** command in `macsec-policy` configuration mode. The timer ranges from 60 to 2,592,000 seconds, the default being OFF.

### Configuration Example

```
Router#configure
Router(config)#macsec-policy test-policy
Router(config-macsec-policy)#sak-rekey-interval 120
Router(config-macsec-policy)#commit
```

### Running Configuration

```
macsec-policy test-policy
  sak-rekey-interval 120
!
```

### Associated Command

**sak-rekey-interval**

## Applying MACsec Configuration on an Interface

The MACsec service configuration is applied to the host-facing interface of a CE router.

### Guidelines for MACsec Interface Configuration

- Configure different keychains for primary and fallback PSKs.

- We do not recommend to update both primary and fallback PSKs simultaneously, because fallback PSK is intended to recover MACsec session on primary key mismatch.

### MACsec PSK Configuration on an Interface

```
Router#configure terminal
Router(config)#interface Te0/3/0/1/4
Router(config-if)#macsec psk-keychain kc policy mac_policy
```

To apply MACsec configuration on a physical interface without the MACsec policy, use the following command:

```
Router(config-if)#macsec psk-keychain script_key_chain2
```

### MACsec Fallback PSK Configuration on an Interface

It is optional to configure a fallback PSK. If a fallback PSK is configured, the fallback PSK along with the primary PSK ensures that the session remains active even if the primary PSK is mismatched, or there is no active key for the primary PSK.

```
Router(config-if)#macsec psk-keychain kc fallback-psk-keychain fallback_kc policy mac_policy
Router(config-if)#commit
```

## Verifying MACsec Encryption on IOS XR

MACsec encryption on IOS XR can be verified by running relevant commands in the Privileged Executive Mode. The verification steps are the same for MACsec encryption on L2VPN or L3VPN network.

To verify if MACsec encryption has been correctly configured, follow these steps.

### SUMMARY STEPS

1. Verify the MACsec policy configuration.
2. Verify the MACsec configuration on the respective interface.
3. Verify whether the interface of the router is peering with its neighbor after MACsec configuration
4. Verify whether the MKA session is secured with MACsec on the respective interface.
5. Verify the MACsec session counter statistics.

### DETAILED STEPS

**Step 1** Verify the MACsec policy configuration.

**Example:**

```
RP/0/RSP0/CPU0:router# show macsec policy mac_policy
```

```
=====
Policy      Cipher      Key-Svr      Window  Conf
name        Suite       Priority     Size    Offset
=====
```



```
mac_policy GCM-AES-XPN-256 0          64      30
```

If the values you see are different from the ones you configured, then check your configuration by running the **show run macsec-policy** command.

**Step 2** Verify the MACsec configuration on the respective interface.

You can verify the MACsec encryption on the configured interface bundle (MPLS network), P2MP interface (VPLS network), or VLAN sub-interface (EoMPLS PW network).

**Example:**

```
RP/0/RSP0/CPU0:router# show macsec mka summary
```

```
NODE: node0_0_CPU0
```

```
=====
Interface      Status      Cipher Suite      KeyChain
=====
```

```
Fo0/0/0/1/0   Secured    GCM-AES-XPN-256   mac_chain
```

```
Total MACSec Sessions : 1
  Secured Sessions : 1
  Pending Sessions : 0
```

```
RP/0/RSP0/CPU0:router# show macsec mka session interface Fo0/0/0/1/0
```

```
=====
Interface      Local-TxSCI      # Peers      Status      Key-Server
=====
```

Interface	Local-TxSCI	# Peers	Status	Key-Server
Fo0/0/0/1/0	d46d.5023.3709/0001	1	Secured	YES

**! If sub-interfaces are configured, the output would be as follows:**

```
RP/0/RSP0/CPU0:router#show macsec mka session interface Fo0/0/0/1/1.8
```

```
=====
Interface      Local-TxSCI      # Peers      Status      Key-Server
=====
```

Interface	Local-TxSCI	# Peers	Status	Key-Server
Fo0/0/0/1/1.8	e0ac.f172.4124/001d	1	Secured	Yes

The **Status** field in the output confirms that the respective interface is **Secured**. If MACsec encryption is not successfully configured, you will see a status such as **Pending** or **Init**.

**Note** In the VPLS network, because of the configuration on a multi-point interface, the number of live peers displayed is more than 1.

Run the **show run macsec-policy** command in the privileged executive mode to troubleshoot the configuration entered.

**Step 3** Verify whether the interface of the router is peering with its neighbor after MACsec configuration

**Example:**

```
RP/0/RSP0/CPU0:router# show macsec mka session
```

```
NODE: node0_0_CPU0
```

```
=====
Interface      Local-TxSCI      # Peers      Status      Key-Server
=====
```





```

SAK Transmit Wait Time : 0s (Not waiting for any peers to respond)
SAK Retire Time       : 0s (No Old SAK to retire)
MKA Policy Name      : mk_xpnl
Key Server Priority   : 0
Replay Window Size   : 1024
Confidentiality Offset : 50
Algorithm Agility     : 80C201
SAK Cipher Suite     : 0080C20001000004 (GCM-AES-XPN-256)
MACsec Capability    : 3 (MACsec Integrity, Confidentiality, & Offset)
MACsec Desired       : YES
# of MACsec Capable Live Peers      : 2
# of MACsec Capable Live Peers Responded : 0
Live Peer List:
-----
MI                MN                Rx-SCI (Peer)          SSCI KS-Priority
-----
AEC899297F5B0BDEF7C9FC67          225 001d.e5e9.blbf/0001    3          0
0A4C49EE5B7401F1BECB7E22          147 001d.e5e9.f329/0001    2          0
Potential Peer List:
-----
MI                MN                Rx-SCI (Peer)          SSCI KS-Priority
-----

```

The **Status** field in the output verifies if the MKA session is secured with MACsec encryption. The output also displays information about the interface and other MACsec parameters.

**Step 5** Verify the MACsec session counter statistics.

**Example:**

```
RP/0/RSP0/CPU0:router# show macsec mka statistics interface Fo0/0/0/1/0
```

```

MKA Statistics for Session on interface (Fo0/0/0/1/0)
=====
Reauthentication Attempts.. 0

CA Statistics
Pairwise CAKs Derived... 0
Pairwise CAK Rekeys..... 0
Group CAKs Generated.... 0
Group CAKs Received..... 0

SA Statistics
SAKs Generated..... 3
SAKs Rekeyed..... 2
SAKs Received..... 0
SAK Responses Received.. 3

MKPDU Statistics
MKPDUs Transmitted..... 5425
"Distributed SAK".. 8
"Distributed CAK".. 0
MKPDUs Validated & Rx... 4932
"Distributed SAK".. 0
"Distributed CAK".. 0

MKA IDB Statistics
MKPDUs Tx Success..... 5425
MKPDUs Tx Fail..... 0
MKPDUS Tx Pkt build fail... 0
MKPDUs Rx CA Not found.... 0
MKPDUs Rx Error..... 0
MKPDUs Rx Success..... 4932

```

```

MKPDU Failures
  MKPDU Rx Validation (ICV)..... 0
  MKPDU Rx Bad Peer MN..... 0
  MKPDU Rx Non-recent Peerlist MN..... 0
  MKPDU Rx Drop SAKUSE, KN mismatch..... 0
  MKPDU Rx Drop SAKUSE, Rx Not Set..... 0
  MKPDU Rx Drop SAKUSE, Key MI mismatch.. 0
  MKPDU Rx Drop SAKUSE, AN Not in Use.... 0
  MKPDU Rx Drop SAKUSE, KS Rx/Tx Not Set. 0

```

```

SAK Failures
  SAK Generation..... 0
  Hash Key Generation..... 0
  SAK Encryption/Wrap..... 0
  SAK Decryption/Unwrap..... 0

```

**! If sub-interfaces are configured, the output would be as follows:**

```
RP/0/RSP0/CPU0:router# show macsec mka statistics interface Fo0/0/0/1/1.8
```

```
MKA Statistics for Session on interface (Fo0/0/0/1/1.8)
```

```

=====
Reauthentication Attempts.. 0
CA Statistics
  Pairwise CAKs Derived... 0
  Pairwise CAK Rekeys..... 0
  Group CAKs Generated.... 0
  Group CAKs Received..... 0
SA Statistics
  SAKs Generated..... 0
  SAKs Rekeyed..... 0
  SAKs Received..... 9
  SAK Responses Received.. 0
MKPDU Statistics
  MKPDUs Transmitted..... 1973
    "Distributed SAK".. 0
    "Distributed CAK".. 0
  MKPDUs Validated & Rx... 1965
    "Distributed SAK".. 9
    "Distributed CAK".. 0
MKA IDB Statistics
  MKPDUs Tx Success..... 1973
  MKPDUs Tx Fail..... 0
  MKPDUS Tx Pkt build fail... 0
  MKPDUs Rx CA Not found.... 0
  MKPDUs Rx Error..... 0
  MKPDUs Rx Success..... 1965

```

**! In a VPLS network with a mulitpoint interface, the output would be as follows:**

```
RP/0/RSP0/CPU0:router# show macsec mka statistics interface FortyGigE0/0/0/1/0.1
```

```
MKA Statistics for Session on interface (Fo0/0/0/1/0.1)
```

```

=====
Reauthentication Attempts.. 0
CA Statistics
  Pairwise CAKs Derived... 0
  Pairwise CAK Rekeys..... 0
  Group CAKs Generated.... 0
  Group CAKs Received..... 0
SA Statistics
  SAKs Generated..... 0

```

```

SAKs Rekeyed..... 0
SAKs Received..... 2
SAK Responses Received.. 0
MKPDU Statistics
MKPDUs Transmitted..... 1608
  "Distributed SAK".. 0
  "Distributed CAK".. 0
MKPDUs Validated & Rx... 406
  "Distributed SAK".. 2
  "Distributed CAK".. 0
MKA IDB Statistics
MKPDUs Tx Success..... 1608
MKPDUs Tx Fail..... 0
MKPDUs Tx Pkt build fail... 0
MKPDUs Rx CA Not found..... 0
MKPDUs Rx Error..... 0
MKPDUs Rx Success..... 1802

```

The counters display the MACsec PDUs transmitted, validated, and received. The output also displays transmission errors, if any.

---

This completes the verification of MACsec encryption on the IOS-XR.

## Verifying MACsec Encryption on ASR 9000

MACsec encryption on the router hardware can be verified by running relevant commands in the Privileged Executive Mode.

To verify if MACsec encryption has been correctly configured, follow these steps.

### SUMMARY STEPS

1. Verify the MACsec encryption and hardware interface descriptor block (IDB) information on the interface.
2. Use the IDB handle retrieved from Step 1 to verify the platform hardware information.
3. Use the Transmitter SA retrieved from Step 2 to verify the MACsec SA information programmed in the hardware.
4. Verify the MACsec Secure Channel (SC) information programmed in the hardware.

### DETAILED STEPS

---

**Step 1** Verify the MACsec encryption and hardware interface descriptor block (IDB) information on the interface.

**Example:**

```
RP/0/RSP0/CPU0:router# show macsec ea idb interface Fo0/0/0/1/0
```

```

IDB Details:
if_sname : Fo0/0/0/1/0
if_handle : 0x3480
Replay window size : 64
Local MAC : 00:1d:e5:e9:aa:39
Rx SC Option(s) : Validate-Frames Replay-Protect
Tx SC Option(s) : Protect-Frames Always-Include-SCI
Security Policy : MUST SECURE

```

```

Sectag offset : 8
VLAN : Outer tag (etype=0x8100, id=1, priority=0, cfi=0): Inner tag (etype=0x8100, id=1, priority=0,
cfi=0)
Rx SC 1
Rx SCI : 001de5e9b1bf0019
Peer MAC : 00:1d:e5:e9:b1:bf
Stale : NO
SAK Data
SAK[0] : ***
SAK Len : 32
HashKey[0] : ***
HashKey Len : 16
Conf offset : 30
Cipher Suite : GCM-AES-XPB-256
CtxSalt[0] : 83 c3 7b ad 7b 6f 63 16 09 8f f3 d2
Rx SA Program Req[0]: 2015 Oct 09 15:20:53.082
Rx SA Program Rsp[0]: 2015 Oct 09 15:20:53.092

Tx SC
Tx SCI : 001de5e9aa39001a
Active AN : 0
Old AN : 255
Next PN : 1, 0, 0, 0
SAK Data
SAK[0] : ***
SAK Len : 32
HashKey[0] : ***
HashKey Len : 16
Conf offset : 30
Cipher Suite : GCM-AES-XPB-256
CtxSalt[0] : 83 c3 7b ae 7b 6f 63 16 09 8f f3 d2
Tx SA Program Req[0]: 2015 Oct 09 15:20:55.053
Tx SA Program Rsp[0]: 2015 Oct 09 15:20:55.064

```

**! When more than 1 RX SA is configured in P2MP networks, the output would be as follows:**

```

RP/0/RSP0/CPU0:router# show macsec ea idb interface FortyGigE0/0/0/1/0.1
IDB Details:
  if_sname           : Fo0/0/0/1/0.1
  if_handle          : 0x2e40
  Replay window size : 1024
  Local MAC          : e0:ac:f1:72:41:23
  Rx SC Option(s)    : Validate-Frames Replay-Protect
  Tx SC Option(s)    : Protect-Frames Always-Include-SCI
  Security Policy    : MUST SECURE
  Sectag offset      : 8
  VLAN               : Outer tag (etype=0x8100, id=1, priority=0, cfi=0)
                   : Inner tag (etype=0x8100, id=1, priority=0, cfi=0)

Rx SC 1
  Rx SCI             : 001de5e9f3290001
  Peer MAC           : 00:1d:e5:e9:f3:29
  Stale              : NO
  SAK Data
    SAK[1]           : ***

    SAK Len          : 32
    HashKey[1]       : ***
    HashKey Len      : 16
    Conf offset      : 50
    Cipher Suite     : GCM-AES-XPB-256
    CtxSalt[1]       : ae ca 99 2b 7f 5b 0b de f7 c9 fc 67

Rx SC 2
  Rx SCI             : 001de5e9b1bf0001
  Peer MAC           : 00:1d:e5:e9:b1:bf

```

```

Stale                : NO
SAK Data
  SAK[1]             : ***

  SAK Len            : 32
  HashKey[1]         : ***
  HashKey Len        : 16
  Conf offset        : 50
  Cipher Suite       : GCM-AES-XPB-256
  CtxSalt[1]         : ae ca 99 2a 7f 5b 0b de f7 c9 fc 67
Tx SC
Tx SCI               : e0acf17241230001
Active AN            : 1
Old AN               : 0
Next PN              : 1, 1, 0, 0
SAK Data
  SAK[1]             : ***

  SAK Len            : 32
  HashKey[1]         : ***
  HashKey Len        : 16
  Conf offset        : 50
  Cipher Suite       : GCM-AES-XPB-256
  CtxSalt[1]         : ae ca 99 28 7f 5b 0b de f7 c9 fc 67

```

The **if\_handle** field provides the IDB instance location.

The **Replay window size** field displays the configured window size.

The **Security Policy** field displays the configured security policy.

The **Local Mac** field displays the MAC address of the router.

The **Peer Mac** field displays the MAC address of the peer. This confirms that a peer relationship has been formed between the two routers.

**Step 2** Use the IDB handle retrieved from Step 1 to verify the platform hardware information.

**Example:**

```

RP/0/RSP0/CPU0:router# show macsec ea platform hardware
idb location 0/0/CPU0 | b 3480

if_handle : 0x00003480
NPPort : 099 [0x063]
LdaPort : 016 [0x010] SerdesPort : 000 [0x000]
NetSoftPort : 061 [0x03d] SysSoftPort : 062 [0x03e]
Active AN : 0x00000000 Idle AN : 0x000000ff
Match-All Tx SA : 0x80010001 Match-All Rx SA : 0x00010001
Match-All Tx Flow : 0x80000003 Match-All Rx Flow : 0x00000003
Bypass Tx SA : 0x80000000 Bypass Rx SA : 0x00000000
Tx SA[0] : 0x80020002 Tx Flow[0] : 0x8000000c
Tx SA[1] : 0xffffffff Tx Flow[1] : 0xffffffff
Tx SA[2] : 0xffffffff Tx Flow[2] : 0xffffffff
Tx SA[3] : 0xffffffff Tx Flow[3] : 0xffffffff
Rx SA[0] : 0x00020002 Rx Flow[0] : 0x0000000c
Rx SA[1] : 0xffffffff Rx Flow[1] : 0xffffffff
Rx SA[2] : 0xffffffff Rx Flow[2] : 0xffffffff
Rx SA[3] : 0xffffffff Rx Flow[3] : 0xffffffff

```

**Step 3** Use the Transmitter SA retrieved from Step 2 to verify the MACsec SA information programmed in the hardware.



**Example:**

```
RP/0/RSP0/CPU0:router# show macsec ea platform hardware sa
0x80020002 interface Fo0/0/0/1/0 location 0/0/CPU0

MACSEC HW SA Details:
Action Type : 0x00000003
Direction : Egress
Dest Port : 0x00000000
Conf Offset : 00000030
Drop Type : 0x00000002
Drop NonResvd : 0x00000000
SA In Use : YES
ConfProtect : YES
IncludeSCI : YES
ProtectFrame : YES
UseEs : NO
UseSCB : NO
SCI : 00 1d e5 e9 aa 39 00 05
Replay Window : 64 MacsecCryptoAlgo : 7
Direction : Egress AN : 0
AES Key Len : 256 X-Packet Number : 0x0000000000000000
CtxSalt : f8d88dc3e1c5e6a94ca2299
```

The output displays the details of the encryption, such as the AES key, the Auth key, and other parameters.

**Step 4** Verify the MACsec Secure Channel (SC) information programmed in the hardware.

**Example:**

```
RP/0/RSP0/CPU0:router# show macsec ea platform hardware msc
interface Fo0/0/0/1/0 location 0/0/CPU0

MACSEC HW Cfg Details:
Mode : 0x5
Counter Clear on Read : 0x0
SA Fail Mask : 0xffff
VlanCounter Update : 0x1
Global SecFail Mask : 0xffffffff
Latency : 0xff
StaticBypass : 0x0
Should secure : 0x0
Global Frame Validation : 0x2
Ctrl Pkt CC Bypass : 0x1
NonCtrl Pkt CC Bypass : 0x1
Sequence Number Threshold : 0xbfffffff8
Sequence Number Threshold 64bit : 0x000002fffffffffd
Non Matching Non Control Pkts Programming
  Untagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
  Tagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
  BadTagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
  KayTagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
Non Matching Control Pkts Programming
  Untagged : Bypass: 0x1 DestPort : 0x2, DropType : 0xffffffff
  Tagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
  BadTagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
  KayTagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
```

This completes the verification of MACsec encryption on the router hardware.

This completes the configuration and verification of MACsec encryption.

## Configuring and Verifying MACsec Encryption as a Service

This section describes how MACsec can be implemented as a service in a L2VPN or L3VPN setup.



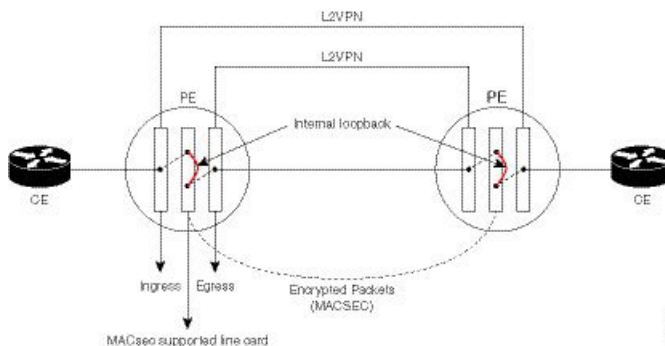
**Note** MACsec encryption is not supported on interface bundles, but is supported on member links .

### Use Case 1: MACsec in an L2VPN Topology

In this topology, MACsec is configured on the PE router (with the interfaces facing the CE router) to provide crypto or encryption service on the PE router as a premium service for selected traffic on the WAN core. The interfaces can be physical ethernet interfaces or VLAN sub-interfaces. The customer can select the traffic that will be part of the encryption.

The following figure illustrates the use of MACsec as a service in an L2VPN network:

**Figure 8: MACsec in an L2VPN topology**



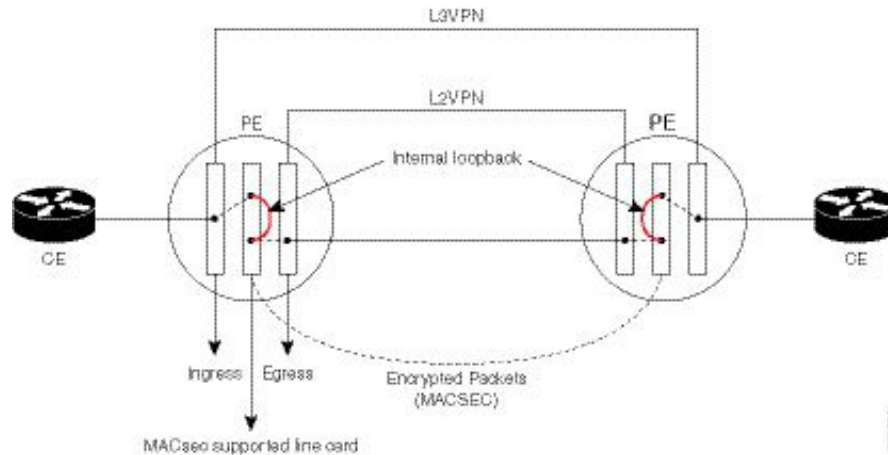
The data transferred between the CE router and the PE router are not encrypted. The data in clear format is sent to the access port of the PE router.

The PE router ports that receive traffic from CE routers divert the traffic using L2 local switching to the line card configured to perform encryption. The MACsec configuration creates internal loopback to the port configured for L2VPN to the opposite PE. After this, the packets are sent completely encrypted to the opposite PE router.

### Use Case 2: MACsec in an L3VPN Topology

The following figure illustrates the use of MACsec as a service in an L3VPN environment. The topology is similar to an L2VPN set up where MACsec is configured on the PE router (where the interfaces facing the CE router) to provide crypto or encryption services on the PE router as a premium service for selected traffic on the WAN core.

Figure 9:



The data transferred between the CE router and the PE router is not encrypted. The data is sent in clear-text format to the PE router access port. The PE router for each sub-interface distinguishes whether the data is part of MACsec encrypted service.

The PE router ports that receive traffic from CE routers divert the traffic using L3 local switching to the line card port configured to do encryption. The MACsec configuration creates internal loopback to the port configured for L2VPN to the opposite PE router. After this, the packets are sent completely encrypted to the opposite PE.

### Restrictions

Ports usage for encryption on the line card must meet the following criteria:

- The ports must be TenGigE interfaces.
- Both the ports must belong either to an A9K-MPA-20X10GE MPA, or they must be breakout interfaces from one of the A9K-8X100GE-SE, A9K-8X100GE-TR, A9K-4X100GE-SE, or A9K-4X100GE-TR line cards.
- If the interfaces belong to A9K-MPA-20x10GE line card, then both the interfaces must be either in port range 0-9, or in port range 10-19. One interface from range 0-9 and other from 10-19 must not be selected.
- If the interfaces are breakout interfaces, then both of them must belong to the same HundredGigE port.



**Note** These restrictions apply only to MACsec interfaces. These restrictions do not apply to the CE or core-facing interfaces.

## Configuring MACsec as a Service

### SUMMARY STEPS

1. Enter interface configuration mode.
2. Configure the MACsec service.

3. Commit your configuration and exit global configuration mode.
4. Confirm the MACsec policy configuration.

## DETAILED STEPS

---

**Step 1** Enter interface configuration mode.

**Example:**

```
RP/0/RSP/CPU0:router# interface <interface> 15.10 l2transport
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 10
```

**Step 2** Configure the MACsec service.

**Example:**

```
RP/0/RSP0/CPU0:router(config-subif)# macsec-service decrypt-port <intf>17.10 psk-keychain
<keychain_name> [policy <macsec_policy>]
```

**Step 3** Commit your configuration and exit global configuration mode.

**Example:**

```
RP/0/RSP0/CPU0:router# commit
RP/0/RSP0/CPU0:router# exit
```

**Step 4** Confirm the MACsec policy configuration.

**Example:**

```
RP/0/RSP0/CPU0:router#
show running-config interface <interface> 15.10

interface <interface> 15.10
 macsec-service decrypt-port <intf>17.10 psk-keychain <keychain_name> [policy <macsec_policy>]
 encapsulation dot1q 10
```

---

## Configuring MACsec Service for L2VPN Network

Configuring the MACsec service for L2VPN network, involves the following steps:

### SUMMARY STEPS

1. Enter global configuration mode.
2. Enter interface configuration mode and configure port facing the CE router.
3. Enable MACsec service.
4. Configure service port.
5. Configure the Xconnect group between ports.

## 6. Connect the ports.

### DETAILED STEPS

---

**Step 1** Enter global configuration mode.

**Example:**

```
RP/0/RSP0/CPU0:router# configure
```

**Step 2** Enter interface configuration mode and configure port facing the CE router.

The interface can be a physical interface or a VLAN sub-interface.

**Example:**

```
RP/0/RSP0/CPU0:router(config)# interface <interface>15.10 l2transport
encapsulation dot1q 10
```

**Step 3** Enable MACsec service.

**Example:**

```
RP/0/RSP0/CPU0:router(config-if)# interface <interface>16.10 l2transport
encapsulation dot1q 10
macsec-service decrypt-port <intf>17.10 psk-keychain <keychain_name> [policy <macsec_policy>]
```

**Step 4** Configure service port.

**Example:**

```
RP/0/RSP0/CPU0:router(config-if)# interface <interface>17.10 l2transport
encapsulation dot1q 10
```

**Step 5** Configure the Xconnect group between ports.

**Example:**

```
RP/0/RSP0/CPU0:router(config-if)# l2vpn
xconnect group local_macsec
p2p local_macsec
interface <interface>15.10
interface <interface>16.10
```

**Step 6** Connect the ports.

**Example:**

```
RP/0/RSP0/CPU0:router(config-if)l2vpn
xconnect group ext_macsec
p2p ext_macsec
interface <interface>17.10
neighbor ipv4 <a.b.c.d> pw-id <num>
!
```

---

# Configuring MACsec Service for L3VPN Network

Configuring the MACsec service for L3VPN network, involves the following steps:

## SUMMARY STEPS

1. Enter global configuration mode.
2. Enter interface configuration mode and configure port facing the CE router
3. Configure the PE1 router with virtual routing details.
4. Enable MACsec service.
5. Configure service port.
6. Configure the Xconnect between ports.
7. Configure ports.
8. Configure OSPF on the core interface.
9. Configure MPLS on the core interface.

## DETAILED STEPS

**Step 1** Enter global configuration mode.

**Example:**

```
RP/0/RSP0/CPU0:router# configure
```

**Step 2** Enter interface configuration mode and configure port facing the CE router

**Example:**

```
RP/0/RSP0/CPU0:router(config-if)# interface TenGigE0/4/0/0.1
ipv4 address 161.1.1.1 255.255.255.0
encapsulation dot1q 1
```

**Step 3** Configure the PE1 router with virtual routing details.

**Example:**

```
RP/0/RSP0/CPU0:router(config-if)# interface TenGigE0/3/0/0/1.1
vrf vrf_1
ipv4 address 161.1.1.2 255.255.255.0
encapsulation dot1q 1
```

**Step 4** Enable MACsec service.

**Example:**

```
RP/0/RSP0/CPU0:router(config-if)# interface TenGigE0/3/0/0/2.1
vrf vrf_1
ipv4 address 181.1.1.1 255.255.255.0
macsec-service decrypt-port TenGigE0/3/0/0/3.1 psk-keychain script_key_chain1
encapsulation dot1q 1
```

**Step 5** Configure service port.

**Example:**

```
RP/0/RSP0/CPU0:router(config-if)#interface TenGigE0/3/0/0/3.1 l2transport
encapsulation dot1q 1
!
```

**Step 6** Configure the Xconnect between ports.

**Example:**

```
RP/0/RSP0/CPU0:router#configure
RP/0/RSP0/CPU0:router(config)#l2vpn
xconnect group l3serv_xc_gp_1
p2p l3serv_xc_p2p_1
interface TenGigE0/3/0/0/3.1
neighbor ipv4 3.3.3.3 pw-id 1
!
!
```

**Step 7** Configure ports.

**Example:**

```
RP/0/RSP0/CPU0:router#(config)
router bgp 100
bgp router-id 2.2.2.2
address-family ipv4 unicast
!
address-family vpnv4 unicast
!
neighbor 3.3.3.3
remote-as 100
update-source Loopback1
address-family vpnv4 unicast
!
!
vrf vrf_1
rd 1234:1
address-family ipv4 unicast
redistribute connected
redistribute static
!
neighbor 181.1.1.2
remote-as 100
address-family ipv4 unicast
!
!
!
```

**Step 8** Configure OSPF on the core interface.

**Example:**

```
RP/0/RSP0/CPU0:router#
macsec-PE1#sh run router ospf
router ospf core
router-id 2.2.2.2
redistribute connected
redistribute static
area 0
interface Loopback1
!
```

```

interface TenGigE0/1/0/1
!
!

```

**Step 9** Configure MPLS on the core interface.

**Example:**

```

RP/0/RSP0/CPU0:router#
mpls ldp
 graceful-restart
 router-id 2.2.2.2
 interface TenGigE0/1/0/1
!
!

```

## Applying MACsec Service Configuration on an Interface

The MACsec service configuration is applied to the host-facing interface of a CE router.

### SUMMARY STEPS

1. Enter the global configuration mode.
2. Enter the interface configuration mode.
3. If you are configuring VLAN sub-interfaces, configure the encapsulation as shown.
4. Apply the MACsec service configuration on an interface.
5. Commit your configuration.

### DETAILED STEPS

**Step 1** Enter the global configuration mode.

**Example:**

```
RP/0/RSP0/CPU0:router# configure
```

**Step 2** Enter the interface configuration mode.

The interface can be a physical interface or a VLAN sub-interface.

**Example:**

```
RP/0/RSP0/CPU0:router(config)# interface Te0/3/0/1/4
```

**Step 3** If you are configuring VLAN sub-interfaces, configure the encapsulation as shown.

**Example:**

**! For 802.1q encapsulation with a single tag**

```
RP/0/RSP0/CPU0:router(config-if)# encapsulation dot1q 5
```

**! For 802.1q encapsulation with double tags**

```
RP/0/RSP0/CPU0:router(config-if)# encapsulation dot1q 3 second-dot1q 4
```

**! For 802.1ad encapsulation with a single tag**



```
RP/0/RSP0/CPU0:router(config-if)# encapsulation dot1ad 5

! For 802.1ad encapsulation with double tags
RP/0/RSP0/CPU0:router(config-if)# encapsulation dot1ad 3 dot1ad 4
```

**Step 4** Apply the MACsec service configuration on an interface.

To apply MACsec service configuration on an interface, use the following configuration.

**Example:**

```
RP/0/RSP0/CPU0:router(config-if)# macsec-service decrypt-port TenGigE0/3/0/1/5 psk-keychain
script_key_chain1 policy mk_xpn_ltag
RP/0/RSP0/CPU0:router(config-if)# exit
```

**Step 5** Commit your configuration.

**Example:**

```
RP/0/RSP0/CPU0:router(config)# commit
```

## Verifying MACsec Encryption on IOS XR

MACsec encryption on IOS XR can be verified by running relevant commands in the Privileged Executive Mode. The verification steps are the same for MACsec encryption on L2VPN or L3VPN network.

To verify if MACsec encryption has been correctly configured, follow these steps.

### SUMMARY STEPS

1. Verify the MACsec policy configuration.
2. Verify the MACsec configuration on the respective interface.
3. Verify whether the interface of the router is peering with its neighbor after MACsec configuration
4. Verify whether the MKA session is secured with MACsec on the respective interface.
5. Verify the MACsec session counter statistics.

### DETAILED STEPS

**Step 1** Verify the MACsec policy configuration.

**Example:**

```
RP/0/RSP0/CPU0:router# show macsec policy mac_policy
```

```
=====
Policy      Cipher      Key-Svr      Window  Conf
name        Suite       Priority     Size    Offset
=====
mac_policy  GCM-AES-XPN-256  0           64      30
```

If the values you see are different from the ones you configured, then check your configuration by running the **show run macsec-policy** command.

**Step 2** Verify the MACsec configuration on the respective interface.

You can verify the MACsec encryption on the configured interface bundle (MPLS network), P2MP interface (VPLS network), or VLAN sub-interface (EoMPLS PW network).

**Example:**

```
RP/0/RSP0/CPU0:router# show macsec mka summary
```

```
NODE: node0_0_CPU0
```

```
=====
Interface      Status      Cipher Suite      KeyChain
=====
```

```
Fo0/0/0/1/0   Secured    GCM-AES-XPB-256   mac_chain
```

```
Total MACSec Sessions : 1
  Secured Sessions : 1
  Pending Sessions : 0
```

```
RP/0/RSP0/CPU0:router# show macsec mka session interface Fo0/0/0/1/0
```

```
=====
Interface      Local-TxSCI      # Peers      Status      Key-Server
=====
```

Interface	Local-TxSCI	# Peers	Status	Key-Server
Fo0/0/0/1/0	d46d.5023.3709/0001	1	Secured	YES

**! If sub-interfaces are configured, the output would be as follows:**

```
RP/0/RSP0/CPU0:router#show macsec mka session interface Fo0/0/0/1/1.8
```

```
=====
Interface      Local-TxSCI      # Peers      Status      Key-Server
=====
```

Interface	Local-TxSCI	# Peers	Status	Key-Server
Fo0/0/0/1/1.8	e0ac.f172.4124/001d	1	Secured	Yes

The **Status** field in the output confirms that the respective interface is **Secured**. If MACsec encryption is not successfully configured, you will see a status such as **Pending** or **Init**.

**Note** In the VPLS network, because of the configuration on a multi-point interface, the number of live peers displayed is more than 1.

Run the **show run macsec-policy** command in the privileged executive mode to troubleshoot the configuration entered.

**Step 3** Verify whether the interface of the router is peering with its neighbor after MACsec configuration

**Example:**

```
RP/0/RSP0/CPU0:router# show macsec mka session
```

```
NODE: node0_0_CPU0
```

```
=====
Interface      Local-TxSCI      # Peers      Status      Key-Server
=====
```





```

SAK Retire Time      : 0s (No Old SAK to retire)
MKA Policy Name     : mk_xpn1
Key Server Priority  : 0
Replay Window Size  : 1024
Confidentiality Offset : 50
Algorithm Agility   : 80C201
SAK Cipher Suite    : 0080C20001000004 (GCM-AES-XPN-256)
MACsec Capability   : 3 (MACsec Integrity, Confidentiality, & Offset)
MACsec Desired      : YES
# of MACsec Capable Live Peers      : 2
# of MACsec Capable Live Peers Responded : 0
Live Peer List:
  MI              MN              Rx-SCI (Peer)          SSCI KS-Priority
  -----
  AEC899297F5B0BDEF7C9FC67          225 001d.e5e9.b1bf/0001    3          0
  0A4C49EE5B7401F1BECB7E22          147 001d.e5e9.f329/0001    2          0
Potential Peer List:
  MI              MN              Rx-SCI (Peer)          SSCI KS-Priority
  -----

```

The **Status** field in the output verifies if the MKA session is secured with MACsec encryption. The output also displays information about the interface and other MACsec parameters.

**Step 5** Verify the MACsec session counter statistics.

**Example:**

```
RP/0/RSP0/CPU0:router# show macsec mka statistics interface Fo0/0/0/1/0
```

```

MKA Statistics for Session on interface (Fo0/0/0/1/0)
=====
Reauthentication Attempts.. 0

CA Statistics
Pairwise CAKs Derived... 0
Pairwise CAK Rekeys..... 0
Group CAKs Generated.... 0
Group CAKs Received..... 0

SA Statistics
SAKs Generated..... 3
SAKs Rekeyed..... 2
SAKs Received..... 0
SAK Responses Received.. 3

MKPDU Statistics
MKPDUs Transmitted..... 5425
"Distributed SAK".. 8
"Distributed CAK".. 0
MKPDUs Validated & Rx... 4932
"Distributed SAK".. 0
"Distributed CAK".. 0

MKA IDB Statistics
MKPDUs Tx Success..... 5425
MKPDUs Tx Fail..... 0
MKPDUS Tx Pkt build fail... 0
MKPDUs Rx CA Not found.... 0
MKPDUs Rx Error..... 0
MKPDUs Rx Success..... 4932

```

```

MKPDU Failures
  MKPDU Rx Validation (ICV)..... 0
  MKPDU Rx Bad Peer MN..... 0
  MKPDU Rx Non-recent Peerlist MN..... 0
  MKPDU Rx Drop SAKUSE, KN mismatch..... 0
  MKPDU Rx Drop SAKUSE, Rx Not Set..... 0
  MKPDU Rx Drop SAKUSE, Key MI mismatch.. 0
  MKPDU Rx Drop SAKUSE, AN Not in Use.... 0
  MKPDU Rx Drop SAKUSE, KS Rx/Tx Not Set. 0

SAK Failures
  SAK Generation..... 0
  Hash Key Generation..... 0
  SAK Encryption/Wrap..... 0
  SAK Decryption/Unwrap..... 0

```

**! If sub-interfaces are configured, the output would be as follows:**

```
RP/0/RSP0/CPU0:router# show macsec mka statistics interface Fo0/0/0/1/1.8
```

```

MKA Statistics for Session on interface (Fo0/0/0/1/1.8)
=====
Reauthentication Attempts.. 0
CA Statistics
  Pairwise CAKs Derived... 0
  Pairwise CAK Rekeys.... 0
  Group CAKs Generated.... 0
  Group CAKs Received.... 0
SA Statistics
  SAKs Generated..... 0
  SAKs Rekeyed..... 0
  SAKs Received..... 9
  SAK Responses Received.. 0
MKPDU Statistics
  MKPDUs Transmitted..... 1973
    "Distributed SAK".. 0
    "Distributed CAK".. 0
  MKPDUs Validated & Rx... 1965
    "Distributed SAK".. 9
    "Distributed CAK".. 0
MKA IDB Statistics
  MKPDUs Tx Success..... 1973
  MKPDUs Tx Fail..... 0
  MKPDUs Tx Pkt build fail... 0
  MKPDUs Rx CA Not found..... 0
  MKPDUs Rx Error..... 0
  MKPDUs Rx Success..... 1965

```

**! In a VPLS network with a mulitpoint interface, the output would be as follows:**

```
RP/0/RSP0/CPU0:router# show macsec mka statistics interface FortyGigE0/0/0/1/0.1
```

```

MKA Statistics for Session on interface (Fo0/0/0/1/0.1)
=====
Reauthentication Attempts.. 0
CA Statistics
  Pairwise CAKs Derived... 0
  Pairwise CAK Rekeys.... 0
  Group CAKs Generated.... 0
  Group CAKs Received.... 0
SA Statistics
  SAKs Generated..... 0
  SAKs Rekeyed..... 0

```

```

SAKs Received..... 2
SAK Responses Received.. 0
MKPDU Statistics
MKPDUs Transmitted..... 1608
  "Distributed SAK".. 0
  "Distributed CAK".. 0
MKPDUs Validated & Rx... 406
  "Distributed SAK".. 2
  "Distributed CAK".. 0
MKA IDB Statistics
MKPDUs Tx Success..... 1608
MKPDUs Tx Fail..... 0
MKPDUs Tx Pkt build fail... 0
MKPDUs Rx CA Not found.... 0
MKPDUs Rx Error..... 0
MKPDUs Rx Success..... 1802

```

The counters display the MACsec PDUs transmitted, validated, and received. The output also displays transmission errors, if any.

---

This completes the verification of MACsec encryption on the IOS-XR.

## Verifying MACsec Encryption on ASR 9000

MACsec encryption on the router hardware can be verified by running relevant commands in the Privileged Executive Mode.

To verify if MACsec encryption has been correctly configured, follow these steps.

### SUMMARY STEPS

1. Verify the MACsec encryption and hardware interface descriptor block (IDB) information on the interface.
2. Use the IDB handle retrieved from Step 1 to verify the platform hardware information.
3. Use the Transmitter SA retrieved from Step 2 to verify the MACsec SA information programmed in the hardware.
4. Verify the MACsec Secure Channel (SC) information programmed in the hardware.

### DETAILED STEPS

**Step 1** Verify the MACsec encryption and hardware interface descriptor block (IDB) information on the interface.

**Example:**

```
RP/0/RSP0/CPU0:router# show macsec ea idb interface Fo0/0/0/1/0
```

```

IDB Details:
if_sname : Fo0/0/0/1/0
if_handle : 0x3480
Replay window size : 64
Local MAC : 00:1d:e5:e9:aa:39
Rx SC Option(s) : Validate-Frames Replay-Protect
Tx SC Option(s) : Protect-Frames Always-Include-SCI
Security Policy : MUST SECURE
Sectag offset : 8

```

```

VLAN : Outer tag (etype=0x8100, id=1, priority=0, cfi=0): Inner tag (etype=0x8100, id=1, priority=0,
cfi=0)
Rx SC 1
Rx SCI : 001de5e9b1bf0019
Peer MAC : 00:1d:e5:e9:b1:bf
Stale : NO
SAK Data
SAK[0] : ***
SAK Len : 32
HashKey[0] : ***
HashKey Len : 16
Conf offset : 30
Cipher Suite : GCM-AES-XPB-256
CtxSalt[0] : 83 c3 7b ad 7b 6f 63 16 09 8f f3 d2
Rx SA Program Req[0]: 2015 Oct 09 15:20:53.082
Rx SA Program Rsp[0]: 2015 Oct 09 15:20:53.092

Tx SC
Tx SCI : 001de5e9aa39001a
Active AN : 0
Old AN : 255
Next PN : 1, 0, 0, 0
SAK Data
SAK[0] : ***
SAK Len : 32
HashKey[0] : ***
HashKey Len : 16
Conf offset : 30
Cipher Suite : GCM-AES-XPB-256
CtxSalt[0] : 83 c3 7b ae 7b 6f 63 16 09 8f f3 d2
Tx SA Program Req[0]: 2015 Oct 09 15:20:55.053
Tx SA Program Rsp[0]: 2015 Oct 09 15:20:55.064

```

**! When more than 1 RX SA is configured in P2MP networks, the output would be as follows:**

```

RP/0/RSP0/CPU0:router# show macsec ea idb interface FortyGigE0/0/0/1/0.1
IDB Details:
  if_sname           : Fo0/0/0/1/0.1
  if_handle          : 0x2e40
  Replay window size : 1024
  Local MAC          : e0:ac:f1:72:41:23
  Rx SC Option(s)   : Validate-Frames Replay-Protect
  Tx SC Option(s)   : Protect-Frames Always-Include-SCI
  Security Policy    : MUST SECURE
  Sectag offset      : 8
  VLAN               : Outer tag (etype=0x8100, id=1, priority=0, cfi=0)
                   : Inner tag (etype=0x8100, id=1, priority=0, cfi=0)

Rx SC 1
  Rx SCI             : 001de5e9f3290001
  Peer MAC           : 00:1d:e5:e9:f3:29
  Stale              : NO
  SAK Data
    SAK[1]           : ***

    SAK Len          : 32
    HashKey[1]       : ***
    HashKey Len      : 16
    Conf offset      : 50
    Cipher Suite     : GCM-AES-XPB-256
    CtxSalt[1]       : ae ca 99 2b 7f 5b 0b de f7 c9 fc 67

Rx SC 2
  Rx SCI             : 001de5e9b1bf0001
  Peer MAC           : 00:1d:e5:e9:b1:bf
  Stale              : NO

```



```

SAK Data
  SAK[1]          : ***

  SAK Len         : 32
  HashKey[1]     : ***
  HashKey Len    : 16
  Conf offset    : 50
  Cipher Suite   : GCM-AES-XPB-256
  CtxSalt[1]    : ae ca 99 2a 7f 5b 0b de f7 c9 fc 67
Tx SC
  Tx SCI         : e0acf17241230001
  Active AN      : 1
  Old AN         : 0
  Next PN       : 1, 1, 0, 0
  SAK Data
    SAK[1]      : ***

    SAK Len     : 32
    HashKey[1]  : ***
    HashKey Len : 16
    Conf offset : 50
    Cipher Suite: GCM-AES-XPB-256
    CtxSalt[1] : ae ca 99 28 7f 5b 0b de f7 c9 fc 67

```

The **if\_handle** field provides the IDB instance location.

The **Replay window size** field displays the configured window size.

The **Security Policy** field displays the configured security policy.

The **Local Mac** field displays the MAC address of the router.

The **Peer Mac** field displays the MAC address of the peer. This confirms that a peer relationship has been formed between the two routers.

**Step 2** Use the IDB handle retrieved from Step 1 to verify the platform hardware information.

**Example:**

```

RP/0/RSP0/CPU0:router# show macsec ea platform hardware
idb location 0/0/CPU0 | b 3480

if_handle : 0x00003480
NPPort : 099 [0x063]
LdaPort : 016 [0x010] SerdesPort : 000 [0x000]
NetSoftPort : 061 [0x03d] SysSoftPort : 062 [0x03e]
Active AN : 0x00000000 Idle AN : 0x000000ff
Match-All Tx SA : 0x80010001 Match-All Rx SA : 0x00010001
Match-All Tx Flow : 0x80000003 Match-All Rx Flow : 0x00000003
Bypass Tx SA : 0x80000000 Bypass Rx SA : 0x00000000
Tx SA[0] : 0x80020002 Tx Flow[0] : 0x8000000c
Tx SA[1] : 0xffffffff Tx Flow[1] : 0xffffffff
Tx SA[2] : 0xffffffff Tx Flow[2] : 0xffffffff
Tx SA[3] : 0xffffffff Tx Flow[3] : 0xffffffff
Rx SA[0] : 0x00020002 Rx Flow[0] : 0x0000000c
Rx SA[1] : 0xffffffff Rx Flow[1] : 0xffffffff
Rx SA[2] : 0xffffffff Rx Flow[2] : 0xffffffff
Rx SA[3] : 0xffffffff Rx Flow[3] : 0xffffffff

```

**Step 3** Use the Transmitter SA retrieved from Step 2 to verify the MACsec SA information programmed in the hardware.

**Example:**

```
RP/0/RSP0/CPU0:router# show macsec ea platform hardware sa
0x80020002 interface Fo0/0/0/1/0 location 0/0/CPU0
```

```
MACSEC HW SA Details:
Action Type : 0x00000003
Direction : Egress
Dest Port : 0x00000000
Conf Offset : 00000030
Drop Type : 0x00000002
Drop NonResvd : 0x00000000
SA In Use : YES
ConfProtect : YES
IncludeSCI : YES
ProtectFrame : YES
UseEs : NO
UseSCB : NO
SCI : 00 1d e5 e9 aa 39 00 05
Replay Window : 64 MacsecCryptoAlgo : 7
Direction : Egress AN : 0
AES Key Len : 256 X-Packet Number : 0x0000000000000000
CtxSalt : f8d88dc3e1c5e6a94ca2299
```

The output displays the details of the encryption, such as the AES key, the Auth key, and other parameters.

**Step 4** Verify the MACsec Secure Channel (SC) information programmed in the hardware.

**Example:**

```
RP/0/RSP0/CPU0:router# show macsec ea platform hardware msc
interface Fo0/0/0/1/0 location 0/0/CPU0
```

```
MACSEC HW Cfg Details:
Mode : 0x5
Counter Clear on Read : 0x0
SA Fail Mask : 0xffff
VlanCounter Update : 0x1
Global SecFail Mask : 0xffffffff
Latency : 0xff
StaticBypass : 0x0
Should secure : 0x0
Global Frame Validation : 0x2
Ctrl Pkt CC Bypass : 0x1
NonCtrl Pkt CC Bypass : 0x1
Sequence Number Threshold : 0xbfffffff8
Sequence Number Threshold 64bit : 0x000002fffffffffd
Non Matching Non Control Pkts Programming
  Untagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
  Tagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
  BadTagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
  KayTagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
Non Matching Control Pkts Programming
  Untagged : Bypass: 0x1 DestPort : 0x2, DropType : 0xffffffff
  Tagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
  BadTagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
  KayTagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
```

---

This completes the verification of MACsec encryption on the router hardware.

This completes the configuration and verification of MACsec encryption.

## Global MACsec Shutdown

The MACsec shutdown feature allows administrator to disable MACsec and re-enable it without modifying the existing MACsec configuration.

Enabling the **macsec shutdown** command, brings down all MACsec sessions on the MACsec-enabled interfaces and resets ports to non-macsec mode. The already existing MACsec configurations remain unaffected by enabling this feature.

Disabling the **macsec shutdown** command, brings up macsec sessions for the configured interfaces and enforces MACsec policy on the port. This feature is disabled by default.

## Configure MACsec Shutdown

The following configuration enables the MACsec shutdown on a chassis:

```
RP/0/RP0/CPU0:router# configure terminal
RP/0/RP0/CPU0:router(config)# macsec shutdown
```



### Warning

Configuring **macsec shutdown** command disables MACsec on all data ports, system wide. Execute **clear** command to erase cached configuration or **commit** command to continue.

## Verify MACsec Shutdown

The **show macsec mka session** command displays a shutdown banner indicating that the MACsec shutdown is enabled.

```
RP/0/RP0/CPU0:router# show macsec mka session
Fri Apr 13 11:56:57.409 IST
```

```
***** MACsec shutdown enabled *****
```

The **show macsec mka interface detail** command displays a shutdown banner and the interface-related information.

```
RP/0/RP0/CPU0:fretta-2#show macsec mka interface detail
Fri Apr 13 11:57:02.685 IST
```

```
***** MACsec shutdown enabled *****
```

```
Number of interfaces on node node0_3_CPU0 : 1
```

```
-----
Interface Name           : HundredGigE0/3/0/8
Interface Namestring     : HundredGigE0/3/0/8
Interface short name     : Hu0/3/0/8
Interface handle         : 0x1800170
Interface number         : 0x1800170
Interface MAC            : 008a.9622.a9d0
Ethertype                : 888E
MACsec Shutdown         : TRUE
Config Received          : TRUE
```

```

IM notify Complete      : TRUE
Interface CAPS Add     : FALSE
RxSA CAPS Add          : FALSE
TxSA CAPS Add          : FALSE
MKA PSK Info
  Key Chain Name       : kc1
  MKA Cipher Suite     : AES-256-CMAC
  CKN                  : 12 34 56
MKA fallback_PSK Info
  fallback keychain Name : fb1
  MKA Cipher Suite     : AES-256-CMAC
  CKN                  : ff ff ff
Policy                  : *DEFAULT POLICY*

```

## Syslog Messages for MACsec Shutdown

The following syslog messages are generated when MACsec shutdown is enabled.

```

%L2-MKA-5-MACSEC_SHUTDOWN_ENABLED : Shutdown ON, disable MACsec on all MACsec enabled ports
%L2-MKA-5-SESSION_STOP             : (Hu0/3/0/8) MKA session stopped,
CKN                                 : 123456
%L2-MKA-4-SESSION_UNSECURED        : (Hu0/3/0/8) MKA Session was stopped and is not secured,
CKN                                 :123456
%L2-MKA-5-MACSEC_DISABLED          : (Hu0/3/0/8), MACsec disabled (shutdown ON)

```

The following syslog messages are generated when MACsec shutdown is disabled.

```

%L2-MKA-5-MACSEC_SHUTDOWN_DISABLED : Shutdown OFF, resume MACsec on all MACsec enabled ports
%L2-MKA-5-MACSEC_ENABLED           : (Hu0/3/0/8), MACsec enabled with MUST_SECURE
%L2-MKA-5-SESSION_START            : (Hu0/3/0/8) MKA session started
CKN                                 : 123456
%L2-MKA-6-MKPDU_ICV_SUCCESS        : (Hu0/3/0/8), ICV verification success for
RxSCI (008a.9600.60b0/0001), CKN(123456)
%L2-MKA-6-FALLBACK_PSK_MKPDU_ICV_SUCCESS : (Hu0/3/0/8), ICV verification success for
RxSCI (008a.9600.60b0/0001), CKN(FFFFFF)
%L2-MKA-5-SESSION_SECURED          : (Hu0/3/0/8) MKA session secured
CKN                                 : 123456

```